

Cloudflare Pages vs Workers - Know the Difference When Setting Up

The Mistake We Made

When creating the project in Cloudflare Dashboard, we accidentally created a **Worker** instead of a **Page**. Workers require a deploy command (`npx wrangler pages deploy`), which failed because it couldn't authenticate inside the build environment. Pages just serves your files — no deploy command needed.

How to Tell the Difference

Workers Setup (WRONG for static sites)

- Asks for a **Deploy command** (required field)
- Asks for a **Non-production branch deploy command**
- Uses ``npx wrangler deploy`` style commands
- Designed for serverless functions and logic at the edge

Pages Setup (RIGHT for static sites)

- Asks for **Build command** (optional, can be blank)
- Asks for **Build output directory** (set to ``/``)
- **No deploy command field at all**
- Designed for hosting static sites (HTML/CSS/JS)

Correct Pages Setup for a Static HTML Site

Setting	Value
Project name	<code>`saltwaterbrc-site`</code>
Production branch	<code>`main`</code>
Framework preset	None
Build command	*(leave blank)*
Build output directory	<code>`/`</code>

That's it. No deploy command, no environment variables, no API tokens.

Where to Create a Pages Project

1. Cloudflare Dashboard → **Workers & Pages** 2. Click **Create** 3. Look for the **Pages** tab — click it 4. Select **Connect to Git** 5. Choose your repo 6. Configure build settings (see table above) 7. Click **Save and Deploy**

The Architecture Difference

Workers: - Runs JavaScript/TypeScript code at the edge - Handles requests, API logic, redirects - Requires a deploy command to push code - Think: serverless functions
Pages: - Hosts static files (HTML, CSS, JS, images) - Serves them from 300+ edge locations - Auto-deploys from GitHub on every push - Think: website hosting

When to Use Each

Use Case	Tool
Host a website	Pages
Blog or content site	Pages
API endpoint	Workers
A/B testing logic	Workers
Form handler	Workers
AI-powered features	Workers
Redirect rules	Workers

Later in this project, we'll use **both** — Pages for the site, Workers for the smart features on top of it.

The Sales Version

"Pages and Workers serve different purposes. Pages is your hosting layer — deploy a site to the global edge with zero config. Workers is your compute layer — run code at the edge for API logic, personalization, AI features. Most customers use both: Pages for the frontend, Workers for the backend logic. No origin server needed for either."